REMARKS

Administrative Overview

Claims 1–19 were initially presented for examination. The present Office Action, mailed on May 1, 2002, objected to the specification and the drawings and rejected the pending claims. Claims 1–14 were rejected under 35 USC §112, ¶2 for indefiniteness. Claims 1, 5, 6, and 11–14 were additionally rejected under 35 USC §103(a) as unpatentable over U.S. Patent No. 5,689,708 to Regnier et al. ("*Regnier*") in view of U.S. Patent No. 5,671,354 to Ito et al. ("*Ito*"). Claims 2, 3, 7–10, and 15–19 were further rejected under 35 USC §103(a) over *Regnier* in further view of *Ito* and U.S. Patent No. 5,860,068 to Cook ("*Cook*"). Claim 4 was additionally rejected over *Regnier* in further view of *Ito* and U.S. Patent No. 5,794,207 to Walker et al. ("*Walker*").

Upon entry of this paper, claims 1–14 and 16–36 are pending; claim 15 is cancelled. Applicants respectfully submit that support for the amendments made can be found throughout the specification and the claims as initially filed.

The Specification Has Been Amended to Correct Matters of Form

The specification was objected to for improper usage of trademarks. Specifically, the specification was objected to for failure to capitalize the trademarks PENTIUM®, MACINTOSH®, WINDOWS®, JAVA®, UNIX®, and ACTIVEX® and include the appropriate generic terminology. Applicants respectfully submit that the foregoing amendments to the specification overcome this objection.

The Drawings Are in Compliance with Rule 1.84

The drawings were objected to for failing to comply with Rule 1.84(p)(5) for purportedly using a reference sign "47" not referred to in the description. Applicants respectfully submit that this objection was issued in error. At pages 15–16, the specification recites, "The client node 10 launches the Program Neighborhood application (e.g., by clicking on the Program Neighborhood icon *47* representing the application)" (emphasis added). Accordingly, Applicants respectfully request that the Examiner withdraw this objection.

The drawings were also objected to for failing to comply with Rule 1.84(p)(4) for using the reference character "41" to designate both the Program Neighborhood icon and Program

Neighborhood application. Applicants respectfully submit that this objection was also issued in error. The text quoted above indicates that the reference number "47" is associated with the Program Neighborhood icon. Similarly, all four occurrences of the reference number "41" in the specification refer to and identify the Program Neighborhood application. See Specification at 15, 16. Therefore, different reference numbers are used to identify different parts of the invention and Rule 1.84(p)(4) is satisfied. Accordingly, Applicants respectfully request that the Examiner also withdraw this objection.

The Claims Have Been Amended to Correct Matters of Form

Claims 1–14 were rejected under 35 USC §112, ¶2 because claims 1, 4, 5, and 6 purportedly lack proper antecedent basis for the limitation "the host system." Applicants respectfully submit that the claims, as amended, overcome this rejection.

The Amended Claims are Patentable over *Regnier* and *Ito*, Taken Alone or In Combination

Claims 1, 5, 6, and 11–14 were rejected under 35 USC §103(a) over *Regnier* in view of *Ito*. Claims 2–4, 7–10, and 15–19 were rejected under 35 USC §103(a) over the combination of *Regnier* and *Ito* with additional references (i.e., *Cook* and *Walker*). Applicants respectfully traverse these rejections.

MPEP 2142 states that, to establish a *prima facie* case of obviousness, it is necessary to show that the prior art references, either alone or in combination, teach or suggest all of the claim limitations. The Applicants respectfully submit that, as amended, independent claims 1 and 16 recite limitations that the cited references, either alone or in combination, neither teach nor suggest; independent claim 15 has been cancelled. The Applicants also note that if an independent claim is nonobvious under 35 USC §103, then by necessity the claims that depend therefrom are likewise nonobvious. See MPEP 2143.03.

As disclosed and claimed, the present invention relates to an apparatus and method for determining the program neighborhood of a client node in a client-server network. See Abstract. To make the client node aware of its program neighborhood, a host server collects application-related information, like information identifying the application, corresponding to application programs hosted by servers in the network. See id. Then, information representing those

application programs that are available to the client node is transmitted from the host server to the client node for display. Accordingly, independent claim 1, as amended, recites a method comprising:

> "(a) receiving, by a web server, an *identification of at least one application program* hosted by at least one of a plurality of servers;
>
> (b) *determining*, at the web server, for a hosted application program *a server selected from the plurality of servers for executing that application program* based on the received information;
>
> (c) *creating, at the web server, a page describing a display of hosted application programs* available to the client system; and
>
> (d) transmitting the created page *to the client system for display.*"

(emphasis added). Independent claim 16, as amended, similarly recites a web server comprising:

> "a service module *collecting an identification of at least one application program* hosted by at least one of the plurality of servers;
>
> a database storing the collected identifications wherein *the service module determines* for at least one application program *a server selected from the plurality of servers for executing the application program* based on the received information;
>
> an output display creation engine *creating a page describing a display* of hosted program available to the client system; and
>
> a transmitter transmitting the created page *to the client system for display.*"

(emphasis added).

That is, Applicants' present invention relates to a web server that collects information identifying applications hosted by a plurality of servers, determines a server for executing the application, creates a page—at the web server—describing a display of the available applications, and provides it to a client system, whereupon the client uses the page to generate a display of the available applications. None of the cited references teach or suggest these limitations. The cited art is either inapposite or teaches a server receiving accounting information from neighboring servers, which is not the invention that the Applicants have disclosed and claimed.

The *Regnier* reference teaches a resource manager in a client/server computer network controlling the availability of system resources. See Abstract. As the Office Action concedes, "Regnier does no [sic] show a host server connected to plurality of servers." See Office Action at 4. Therefore, *Ito* is relied upon to teach "[a] host server receiving account information from the plurality of servers." See id.

While this may be true, it is not relevant to the limitations in the claims, as amended. *Ito* does not teach a web server *collecting an identification of at least one application program hosted by at least one of a plurality of servers*. Instead, the *Ito* reference describes a server that receives account balance information from neighboring servers:

> "Normally, the account totalizing program 16 is in a state waiting for information coming from a server (step 140). . . . Then, the program 16 determines whether the *actual account* for the user has exceeded that *budgetary account* (step 142). If the actual account has been found to be excessive, the program 16 notifies each server to prohibit the access to that server from that user (step 143) and enters into the wait state again"

(emphasis added). See col. 7, ln. 45–46, 54–59. This is not a web server collecting information identifying hosted applications. Instead, it is an illustration of how accounting information stored on various servers can be accessed by a central server.

Ito also fails to meet the other claim limitations in that it fails to teach or suggest the determination of a server for executing an application, the creation of a page at the server describing the available applications, or the transmission of the created page to the client for display.

Therefore, Applicants respectfully submit that amended independent claims 1 and 16, and the remaining claims that depend thereon, are patentably distinct over *Regnier* and *Ito* taken separately or in combination, which neither teach nor suggest Applicants' claimed invention.

New Claims 20–35 are Patentable over the Cited References, Taken Alone or In Combination

The new claims consist of independent claim 23, claims that depend therefrom, and claims that depend from independent claims 1 and 16. Applicants respectfully submit that the

new claims that depend from independent claims 1 and 16 are allowable for at least the reasons presented above.

Independent claim 23 relates to an embodiment of the present invention that includes computer-readable means for executing the claimed invention and recites:

> "computer-readable means for receiving, by a web server, an identification of at least one application program hosted by at least one of a plurality of servers;
>
> computer-readable means for determining, at the web server, for a hosted application program, a server selected from the plurality of servers for executing that application program based on the received information;
>
> computer-readable means for creating, at the web server, a page describing a display of hosted application programs available to the client system; and
>
> computer-readable means for transmitting the created page to the client system for display."

As stated above neither *Regnier* nor *Ito*, taken singly or in combination, teaches these features. Applicants respectfully submit that new independent claim 23, and the remaining claims that depend thereon, are patentably distinct from the cited art either separately or in combination, which neither teaches nor suggests Applicants' claimed invention.

CONCLUSION

For these reasons Applicants respectfully submit that independent claims 1 and 16, as amended; claims 2–14 and 17–19, which depend therefrom; and new claims 20–36 are patentably distinct over the cited references, either taken singly or in combination, which neither teach nor suggest Applicants' claimed invention.

In light of the foregoing, we submit that all claims are now in condition for allowance. If the Examiner believes that a telephone conversation with the Applicants' attorney would expedite the allowance of this application, the Examiner is invited to call the undersigned.

Respectfully submitted,

Robert S. Blasi, Esq.
Attorney for Applicants
Testa, Hurwitz, & Thibeault, LLP
High Street Tower
125 High Street
Boston, MA 02110

Date: September 3, 2002
Reg. No. 50,389

Tel. No.: (617) 248-7097
Fax No.: (617) 248-7100

2427859

MARKED-UP COPY OF AMENDED CLAIMS

1. (Once amended) In a network including a client system and a plurality of servers, the plurality of servers including a [host]web server, [the plurality of servers hosting application programs,] a method for presenting to the client system application programs that are available for use, the method comprising [the steps of]:

(a) receiving, by a [host]web server, [application-related information corresponding to]an identification of at least one application program[s] hosted by at least one of a plurality of servers;

[(b) receiving by the host system user credentials from a client system;]

([c]b) determining, at the web server, for [each]a hosted application program a server selected from the plurality of servers for executing that application program based on the received information[whether that hosted application program is available to the client system based on the user credentials and the application-related information];

([d]c) creating, at the web server, [an output]a page describing a display of[indicating each] hosted application programs [that is] available to the client system [for execution]; and

([e]d) transmitting the created [output display]page to the client system for display.

2. (Once amended) The method of claim 1 [wherein step (a) further comprises receiving by a web server application-related information corresponding to application programs hosted by at least one of a plurality of servers]further comprising receiving, by the web server, user credentials from a client system.

3. (Once amended) The method of claim [2]1 wherein step (a) further compris[ing]es [the steps of]:

(a) parsing an SGML document; and

(b) retrieving, responsive to the parsed SGML document, [application-related information corresponding to]an identification of at least one application program[s] hosted by at least one of [a]the plurality of servers.

4. (Once amended) The method of claim [1]2 wherein [step (b)] receiving user credentials comprises receiving by the [host system]web server biometric user credentials from a client system.

5. (Once amended) The method of claim [1]2 wherein [step (b)] receiving user credentials comprises receiving by the [host system]web server encrypted user credentials from a client system.

6. (Once amended) The method of claim [1]2 wherein [step (b)] receiving user credentials further comprises:

([b-]a) receiving by the [host system]web server user credentials from a client system;

([b-]b) authenticating at the [host] web server the user of the client system based on the received user credentials; and

([b-]c) executing a selected one of the available application programs hosted by one of the plurality of servers without requiring further receipt of user credentials from the client system.

7. (Once amended) The method of claim 1 wherein step ([d]c) comprises creating an SGML [output display]page indicating [each] hosted application programs [that is] available to the client system [for execution].

8. (Once amended) The method of claim 1 wherein step ([d]c) comprises creating an HTML [output display]page indicating [each] hosted application programs [that is] available to the client system [for execution].

9. (Once amended) The method of claim 1 wherein step ([d]c) comprises creating an output page describing[display representing] the available application programs as icons in a graphical user interface window.

10. (Once amended) The method of claim 1 wherein step ([e]d) comprises transmitting the created [output display]page to the client system using HTTP.

11. (Once amended) The method of claim 1 further comprising:

([f]e) receiving a request to execute [one of] the [available] hosted application programs; and

([h]f) executing the requested application program at the determined server.

12. (Once amended) The method of claim 11 wherein step ([h]f) further comprises executing the requested application in a window [contained in the output display page]described in the created page.

13. (Once amended) The method of claim 11 further comprising establishing a connection between the client system and the server [hosting]executing the requested application.

14. (Once amended) The method of claim 1 further comprising communicating, by the [host]web server, with the plurality of servers [in order] to determine the application programs hosted by the plurality of servers.

16. (Once amended) In a network including a client system and a plurality of servers hosting applications, a web server comprising:

a service module collecting [application-related information corresponding to]an identification of at least one application program[s] hosted by at least one of the plurality of servers;

[a receiver receiving user credentials from the client system;]

a database storing the collected [application-related information]identifications wherein the service module determines for [each]at least one application program a server selected from the plurality of servers for executing the application program based on the received information[hosted by the plurality of servers whether that hosted application program is available to the client system for execution based on the user credentials and the application-related information stored in the database];

an output display creation engine creating [output displays]a page describing a display of [indicating each] hosted programs available to the client system [for execution]; and

a transmitter transmitting the created [output displays]page to the client system for display.

17. (Once amended) The web server of claim 16 wherein said service module transmits a datagram to at least one of the plurality of servers to collect the [application-related information corresponding to the]identification of at least one application program[s] hosted by those servers.

18. (Once amended) The web server of claim 16 wherein said output display creation engine [parses]creates the page using SGML document templates.

19. (Once amended) The web server of claim 18 wherein said transmitter transmits [available application information]the created page using HTTP.

MARKED-UP COPY OF REPLACEMENT SPECIFICATION PARAGRAPHS

At page 9:

The client node 10 can be any personal computer (e.g., 286, 386, 486, [Pentium, Pentium] PENTIUM, PENTIUM II, [Macintosh] MACINTOSH computer), Windows-based terminal, Network Computer, wireless device, information appliance, RISC Power PC, X-device, workstation, mini computer, main frame computer or other computing device that has a windows-based desktop and sufficient persistent storage for executing application programs downloaded from the application servers 30, 32, 34 across the network 40. Windows-oriented platforms supported by the client node 10 can include [Windows]WINDOWS 3.x, [Windows]WINDOWS 95, [Windows]WINDOWS 98, [Windows]WINDOWS NT 3.51, [Windows]WINDOWS NT 4.0, [Windows]WINDOWS CE, [Macintosh]MACINTOSH, [Java]JAVA, and [Unix]UNIX. The client node 10 can include a display screen 12, a keyboard 14, memory 16 for storing downloaded application programs, a processor 17, and a mouse 18. The memory 16 can provide persistent or volatile storage. The processor 17 can execute the application programs locally on the client node 10 and display a resulting windows-based desktop on the display screen 12. Such local processing on the client node 10 is according to the above-described client-based computing model.

Alternatively, the client node 20 can be any terminal (windows or non-windows based), or thin-client device operating according to a server-based computing model. In a server-based computing model, the execution of application programs occurs entirely on the application servers 30, 32, 34, and the user interface, keystrokes, and mouse movements are transmitted over the network 40 to the client node 20. The user interface can be text driven (e.g., DOS) or graphically driven (e.g., [Windows]WINDOWS). Platforms that can be supported by the client node 20 include DOS and [Windows]WINDOWS CE for windows-based terminals. The client node 20 includes a display screen 22, a keyboard 24, a mouse 28, a processor (not shown), and persistent storage (not shown).

At page 31:

The PNAPI 52 provides automatic and manual management for Program Neighborhood application objects stored in the local cache 60. The local cache 60 can either be refreshed manually by the user of the client node 10, or at a user-definable refresh rate, or by the server at any time during a connection. In a [Windows]WINDOWS implementation, the PNAPI 52 can build remote application file associations and manage the "Start" menu and desktop icons for application object shortcuts.

At page 34:

To run the Program Neighborhood application in a server-based implementation, the user of the client node 20 connects to an initial desktop (at the server 30') and launches the Program Neighborhood application from within that desktop environment. The connection to the initial desktop can occur automatically, e.g., via a logon script of the client node 20, via an entry in the StartUp group in [Windows]WINDOWS 95, or by another centrally managed server specific mechanism. All remote application management and launching is accomplished through this initial desktop.

At page 35:

In one embodiment, the web-based Program Neighborhood application includes a group of objects that manage various aspects of the application. In one embodiment, the application includes three primary object classes that "plug in" to a web server: a gateway object class; a credentials object class; and an applications object class. In some specific embodiments, the object classes are provided as [Java beans]JAVA BEANS. The three primary object classes facilitate: validation of user credentials into a server farm; generation of lists of published applications that a specified user may access; provision of detailed information about a specific published application; and conversion of published application information into an ICA-compatible format.

When provided as [Java beans]JAVA BEANS, the objects can be accessed in a number of different ways. For example, they may be compiled as COM objects and made available to the web server as [ActiveX]ACTIVEX components. In another embodiment, the [Java beans]JAVA

BEANS can be used in their native form, such as when the server uses [Java] JAVA Server Pages technology. In yet another embodiment, the [Java beans]JAVA BEANS can be instantiated and used directly in a [Java]JAVA servlet. In still another embodiment, the server 30 can instantiate the [Java beans]JAVA BEANS as COM objects directly.

At page 36:

A gateway object class handles communications with a target server farm. In one embodiment, the gateway object class is provided as an abstract [Java]JAVA class that cannot be instantiated. A particular gateway object may retrieve application information by communicating with a server farm using a particular protocol, reading cached application information, a combination of these two methods, or other various methods.